LAB # 12

IMPLEMENTATION OF STRUCTURES IN C++

Structures

Structures in C++ is a collection of variables. Structures in C++ can be declared even without the keyword "struct". By default all the members of a structure are "public", even "private" members can also be declared in a function.

Syntax:

```
struct struct-type-name{
type name1: length;
type name2: length;
.
.
type nameN : length;
}variable_list;
```

Example :

```
#include <iostream.h>
struct Emp
{
int empno;
int empsal;
};
void main( )
{
Emp emp1= { 23, 12000};
cout << "Employee Number::" <<
emp1.empno << '\n';
cout << "Employee Salary:: "<< emp1.empsal;
}
```

Result :

Employee Number:: 23
Employee Salary:: 12000
In the above example, the structure "Emp" is used initialize the integers, that are referenced in the "main()" function.

Unions

Unions in C++ is a user defined data type that uses the same memory as other objects from a list of objects. At an instance it contains only a single object.

Syntax:

union union-type-
name{type member-
name; type member-
name;
}union-variables;

Example :

```
#include <iostream.h> union Emp
{
int num;
double sal;
}; int main()
{
Emp value;
value.num = 2;
cout << "Employee Number::" << value.num
<< "\nSalary is:: " << value.sal << endl;
value.sal = 2000.0;
cout << "Employee Number::" << value.num
<< "\nSalary is:: " << value.sal << endl;
return 0;
}
```

Result :

```
Employee number is::2
Salary is::2.122e-314
Employee number is::0
Salary is::2000
```

In the above example, only "value.num" is assigned, but still the "val.sal" gets a value automatically, since the memory locations are same.

Example of comparing size of union and structure

```
#include<iostream.h>
struct Employee1
 {
        int Id;
        char Name[25];
        long Salary;
 };
 union Employee2
 {
        int Id;
        char Name[25];
        long Salary;
 };
  void main()
  {
       cout << "\nSize of Employee1 is : " << sizeof(Employee1);
       cout << "\nSize of Employee2 is : " << sizeof(Employee2);
  }
```

Output :

```
Size of Employee1 is : 31
Size of Employee2 is : 25
```

Array of Structure

Structure is collection of different data type. An object of structure represents a single record in memory, if we want more than one record of structure type, we have to create an array of structure or object. As we know, an array is a collection of similar type, therefore an array can be of structure type.

Syntax for declaring structure array

```
struct struct-name
{
    datatype var1;
    datatype var2;
    -------
    -------
```

```
        datatype varN;
    };
    struct-name obj [ size ];
```

Example for declaring structure array

```
#include<iostream.h>
struct Employee
{
    int Id;
    char Name[25];
    int Age;
    long Salary;
};
void main()
{
    int i;
    Employee Emp[ 3 ];      //Statement 1
    for(i=0;i<2;i++)
    {
        cout << "\nEnter details of " << i+1 << "
        Employee"; cout << "\n\tEnter Employee Id : "; cin
        >> Emp[i].Id;
        cout << "\n\tEnter Employee Name : ";
        cin >> Emp[i].Name;
        cout << "\n\tEnter Employee Age : ";
        cin >> Emp[i].Age;
        cout << "\n\tEnter Employee Salary : ";
        cin >> Emp[i].Salary;
    }
    cout << "\nDetails of Employees";
    for(i=0;i<2;i++)
    cout << "\n"<< Emp[i].Id <<"\t"<< Emp[i].Name <<"\t"
          << Emp[i].Age <<"\t"<< Emp[i].Salary;
  }
```

Output :

```
Enter details of 1 Employee
        Enter Employee Id : 101
        Enter Employee Name :
        SamiEnter Employee Age :
        29
        Enter Employee Salary : 45000
```

```
Enter details of 2 Employee
        Enter Employee Id : 102
        Enter Employee Name :
        AhmedEnter Employee Age :
        31
        Enter Employee Salary : 51000

Details of Employees
          101    Sami      29     45000
          102    Ahmed     31     51000
```

In the above example, we are getting and displaying the data of 3 employee using array of object. Statement 1 is creating an array of Employee Emp to store the records of 2 employees.

Structures as Function Arguments

You can pass a structure as a function argument in very similar way as you pass any other variable or pointer. You would access structure variables in the similar way as you have accessed in the above example −

```
#include <iostream>
#include <cstring>

using namespace std;
void printBook( struct Books book);
struct Books
{
char title[50];
char author[50];
char subject[100];
int book_id;
};


int main()
{
struct Books Book1;
struct Books Book2;


// Declare Book1 of type Book
// Declare Book2 of type Book
// book 1 specification
```

```
strcpy( Book1.title, "Learn C++ Programming");
strcpy( Book1.author, "Chand Miyan");
strcpy( Book1.subject, "C++ Programming");
Book1.book_id = 6495407;

// book 2 specification
strcpy( Book2.title, "Telecom Billing");
strcpy( Book2.author, "Yakit Singha");
strcpy( Book2.subject, "Telecom");
Book2.book_id = 6495700;

// Print Book1 info printBook( Book1 );

// Print Book2 info printBook( Book2 );

return 0;
}

void printBook( struct Books book )
{
cout << "Book title : " << book.title <<endl;
cout << "Book author : " << book.author <<endl;
cout << "Book subject : " << book.subject <<endl;
cout << "Book id : " << book.book_id <<endl;
}
```

When the above code is compiled and executed, it produces the following result −

Book title : Learn C++ Programming
Book author : Chand Miyan
Book subject : C++ Programming
Book id : 6495407
Book title : Telecom Billing
Book author : Yakit Singha
Book subject : Telecom
Book id : 6495700

Lab Tasks

1- Tasks

Create a structure to specify data of customers in a bank. The data to be stored is: Account number, Name, Balance in account. Assume maximum of 200 customers in the bank. Write a function to print the Account number and name of each customer with balance below Rs. 1000.

2- Tasks

Write a menu driven program that depicts the working of a library. The menu options should be:

a) Add book information
b) Display book information
c) List all books of given author
d) List the title of specified book
e) List the count of books in the library
f) List the books in the order of accession number
g) Exit

Create a structure called library to hold accession number, title of the book, author name, price of the book, and flag indicating whether book is issued or not.

3- Tasks

i- Declare a structure named employee that stores the employee id, name, salary and department.

ii- Declare an array of 5 employees for the structure defined in part(i) . Also write statements to assign the following values to the employee [3].
Employee id = "Your_roll_no" salary = 30,000 and department = "IT dept"

iii- Write necessary statement to initialize all the elements of above array.

iv- Write a function to take input in above array of struct employee.

v- Write a function that prints the highest salaried person amongst the employees.

vi- Write a function that search & display records of all those employees, whose salary in greater than 15000.

vii- Write a function that search & display records of all those employees, who are working in Finance department.